SDYZ001

Overview of IEEE Std 91-1984

1984

Explanation of Logic Symbols

Texas Instruments

Overview of IEEE Standard 91-1984

Explanation of Logic Symbols

F.A. Mann
Semiconductor Group

Texas
Instruments

If you have questions on this Explanation of Logic Symbols, please contact:

Texas Instruments Incorporated
F.A. Mann, MS 49
P.O. Box 225012
Dallas, Texas 75265

Telephone (214) 995-2867

IEEE Standards may be purchased from:

Institute of Electrical and Electronics Engineers, Inc.
IEEE Standards Office
345 East 47th Street
New York, N.Y. 10017

International Electrotechnical Commission (IEC) publications may be purchased from:

American National Standards Institute, Inc.
1430 Broadway
New York, N.Y. 10018

Copyright © 1984, Texas Instruments Incorporated

Contents

Title *Page*

1.0 INTRODUCTION 1
2.0 SYMBOL COMPOSITION 1
3.0 QUALIFYING SYMBOLS 3
    3.1 General Qualifying Symbols 3
    3.2 Qualifying Symbols for Inputs and Outputs 3
    3.3 Symbols Inside the Outline 7
4.0 DEPENDENCY NOTATION 7
    4.1 General Explanation 7
    4.2 G, AND 8
    4.3 Conventions for the Application of Dependency Notation in General 9
    4.4 V, OR 10
    4.5 N, Negate (Exclusive-OR) 10
    4.6 Z, Interconnection 11
    4.7 X, Transmission 12
    4.8 C, Control 13
    4.9 S, Set and R, Reset 14
    4.10 EN, Enable 14
    4.11 M, Mode 15
    4.12 A, Address 17
5.0 BISTABLE ELEMENTS 19
6.0 CODERS 20
7.0 USE OF A CODER TO PRODUCE AFFECTING INPUTS 22
8.0 USE OF BINARY GROUPING TO PRODUCE AFFECTING INPUTS 22
9.0 SEQUENCE OF INPUT LABELS 23
10.0 SEQUENCE OF OUTPUT LABELS 24

List of Tables

Table *Title* *Page*

I General Qualifying Symbols 4
II Qualifying Symbols for Inputs and Outputs 5
III Symbols Inside the Outline 6
IV Summary of Dependency Notation 19

List of Illustrations

Figure *Title* *Page*

1 Symbol Composition 2
2 Common-Control Block 2
3 Common-Output Element 3
4 G Dependency Between Inputs 8
5 G Dependency Between Outputs and Inputs 9
6 G Dependency with a Dynamic Input 9
7 ORed Affecting Inputs 9
8 Substitution for Numbers 10
9 V (OR) Dependency 10
10 N (Negate/Exclusive-OR) Dependency 11
11 Z (Interconnection) Dependency 11
12 X (Transmission) Dependency 12
13 CMOS Transmission Gate Symbol and Schematic 12
14 Analog Data Selector (Multiplexer/Demultiplexer) 12
15 C (Control) Dependency 13
16 S (Set) and R (Reset) Dependencies 14
17 EN (Enable) Dependency 15
18 M (Mode) Dependency Affecting Inputs 16
19 Type of Output Determined by Mode 16
20 An Output of the Common-Control Block 17
21 Determining an Output's Function 17
22 Dependent Relationships Affected by Mode 17
23 A (Address) Dependency 18
24 Array of 16 Sections of Four Transparent Latches with 3-State Outputs Comprising a 16-Word × 4-Bit Random-Access Memory 19
25 Four Types of Bistable Circuits 20
26 Coder General Symbol 20
27 An X/Y Code Converter 21
28 AN X/Octal Code Converter 22
29 Producing Various Types of Dependencies 22
30 Producing One Type of Dependency 22
31 Use of Binary Grouping Symbol 23
32 Input Labels 23
33 Factoring Input Labels 24
34 Placement of 3-State Symbols 24
35 Output Labels 24
36 Factoring Output Labels 25

1.0 INTRODUCTION

The International Electrotechnical Commission (IEC) has been developing a very powerful symbolic language that can show the relationship of each input of a digital logic circuit to each output without showing explicitly the internal logic. At the heart of the system is dependency notation, which will be explained in Section 4.

The system was introduced in the USA in a rudimentary form in IEEE/ANSI Standard Y32.14-1973. Lacking at that time a complete development of dependency notation, it offered little more than a substitution of rectangular shapes for the familiar distinctive shapes for representing the basic functions of AND, OR, negation, etc. This is no longer the case.

Internationally, Working Group 2 of IEC Technical Committee TC-3 has prepared a new document (Publication 617-12) that consolidates the original work started in the mid 1960's and published in 1972 (Publication 117-15) and the amendments and supplements that have followed. Similarly for the USA, IEEE Committee SCC 11.9 has revised the publication IEEE Std 91/ANSI Y32.14. Now numbered simply IEEE Std 91-1984, the IEEE standard contains all of the IEC work that has been approved, and also a small amount of material still under international consideration. Texas Instruments is participating in the work of both organizations and this document introduces new logic symbols in accordance with the new standards. When changes are made as the standards develop, future editions will take those changes into account.

The following explanation of the new symbolic language is necessarily brief and greatly condensed from what the standards publications will contain. This is not intended to be sufficient for those people who will be developing symbols for new devices. It is primarily intended to make possible the understanding of the symbols used in various data books and the comparison of the symbols with logic diagrams, functional block diagrams, and/or function tables will further help that understanding.

2.0 SYMBOL COMPOSITION

A symbol comprises an outline or a combination of outlines together with one or more qualifying symbols. The shape of the symbols is not significant. As shown in Figure 1, general qualifying symbols are used to tell exactly what logical operation is performed by the elements. Table I shows general qualifying symbols defined in the new standards. Input lines are placed on the left and output lines are placed on the right. When an exception is made to that convention, the direction of signal flow is indicated by an arrow as shown in Figure 11.

All outputs of a single, unsubdivided element always have identical internal logic states determined by the function of the element except when otherwise indicated by an associated qualifying symbol or label inside the element.

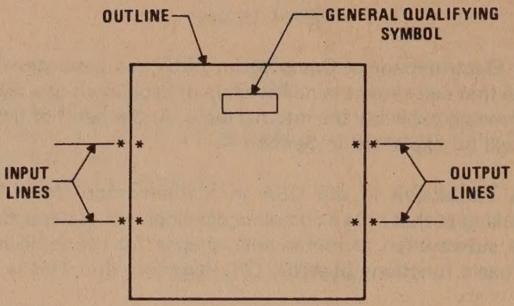

OUTLINE
GENERAL QUALIFYING SYMBOL
INPUT LINES
OUTPUT LINES
*
*Possible positions for qualifying symbols relating to inputs and outputs

Figure 1. Symbol Composition

The outlines of elements may be abutted or embedded in which case the following conventions apply. There is no logic connection between the elements when the line common to their outlines is in the direction of signal flow. There is at least one logic connection between the elements when the line common to their outlines is perpendicular to the direction of signal flow. The number of logic connections between elements will be clarified by the use of qualifying symbols and this is discussed further under that topic. If no indications are shown on either side of the common line, it is assumed there is only one connection.

When a circuit has one or more inputs that are common to more than one element of the circuit, the common-control block may be used. This is the only distinctively shaped outline used in the IEC system. Figure 2 shows that unless otherwise qualified by dependency notation, an input to the common-control block is an input to each of the elements below the common-control block.

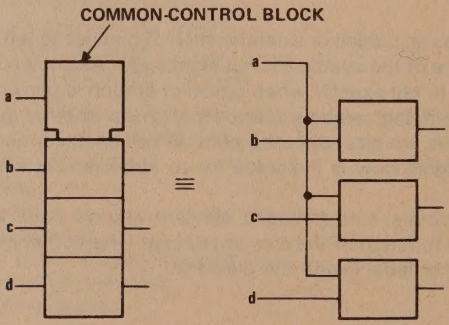


Figure 2. Common-Control Block

A common output depending on all elements of the array can be shown as the output of a common-output element. Its distinctive visual feature is the double line at its top. In addition the common-output element may have other inputs as shown in Figure 3. The function of the common-output element must be shown by use of a general qualifying symbol.

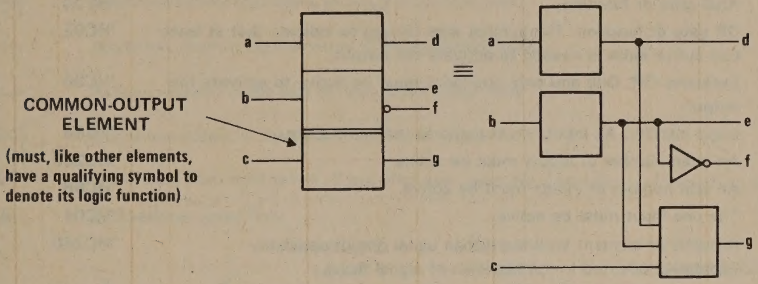

COMMON-OUTPUT ELEMENT
(must, like other elements, have a qualifying symbol to denote its logic function)
a
b
c
d
e
f
g
≡
a
b
c
d
e
f
g

Figure 3. Common-Output Element

3.0 QUALIFYING SYMBOLS

3.1 General Qualifying Symbols

Table I shows general qualifying symbols defined by IEEE Standard 91. These characters are placed near the top center or the geometric center of a symbol or symbol element to define the basic function of the device represented by the symbol or of the element.

3.2 Qualifying Symbols for Inputs and Outputs

Qualifying symbols for inputs and outputs are shown in Table II and will be familiar to most users with the possible exception of the logic polarity and analog signal indicators. The older logic negation indicator means that the external 0 state produces the internal 1 state. The internal 1 state means the active state. Logic negation may be used in pure logic diagrams; in order to tie the external 1 and 0 logic states to the levels H (high) and L (low), a statement of whether positive logic (1 = H, 0 = L) or negative logic (1 = L, 0 = H) is being used is required or must be assumed. Logic polarity indicators eliminate the need for calling out the logic convention and are used in various data books in the symbology for actual devices. The presence of the triangular polarity indicator indicates that the L logic level will produce the internal 1 state (the active state) or that, in the case of an output, the internal 1 state will produce the external L level. Note how the active direction of transition for a dynamic input is indicated in positive logic, negative logic, and with polarity indication.

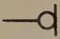
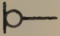
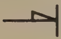
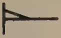
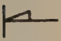
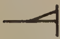
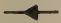
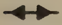
The internal connections between logic elements abutted together in a symbol may be indicated by the symbols shown in Table II. Each logic connection may be shown by the presence of qualifying symbols at one or both sides of the common line and if confusion can arise about the numbers of connections, use can be made of one of the internal connection symbols.

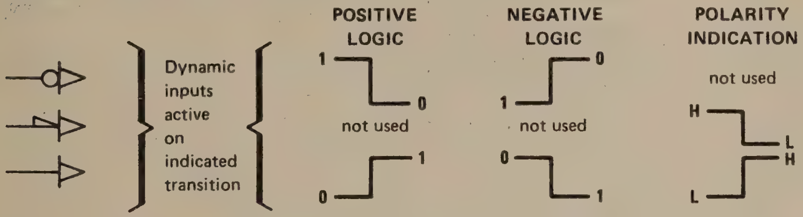
Table I. General Qualifying Symbols

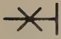
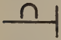
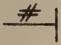
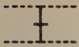
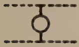
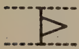
SYMBOL	DESCRIPTION	CMOS EXAMPLE	TTL EXAMPLE
&	AND gate or function.	'HC00	SN7400
≥1	OR gate or function. The symbol was chosen to indicate that at least one active input is needed to activate the output.	'HC02	SN7402
=1	Exclusive OR. One and only one input must be active to activate the output.	'HC86	SN7486
=	Logic identity. All inputs must stand at the same state.	'HC86	SN74180
2k	An even number of inputs must be active.	'HC280	SN74180
2k+1	An odd number of inputs must be active.	'HC86	SN74ALS86
1	The one input must be active.	'HC04	SN7404
▷ or ◁	A buffer or element with more than usual output capability (symbol is oriented in the direction of signal flow).	'HC240	SN74S436
⎎	Schmitt trigger; element with hysteresis.	'HC132	SN74LS18
X/Y	Coder, code converter (DEC/BCD, BIN/OUT, BIN/7-SEG, etc.).	'HC42	SN74LS347
MUX	Multiplexer/data selector.	'HC151	SN74150
DMUX or DX	Demultiplexer.	'HC138	SN74138
Σ	Adder.	'HC283	SN74LS385
P−Q	Subtracter.	*	SN74LS385
CPG	Look-ahead carry generator	'HC182	SN74182
π	Multiplier.	*	SN74LS384
COMP	Magnitude comparator.	'HC85	SN74LS682
ALU	Arithmetic logic unit.	'HC181	SN74LS381
⎍	Retriggerable monostable.	'HC123	SN74LS422
1⎍	Nonretriggerable monostable (one-shot)	'HC221	SN74121
G ⎍⎍	Astable element. Showing waveform is optional.	*	SN74LS320
!G ⎍⎍	Synchronously starting astable.	*	SN74LS624
G! ⎍⎍	Astable element that stops with a completed pulse.	*	*
SRGm	Shift register. m = number of bits.	'HC164	SN74LS595
CTRm	Counter. m = number of bits; cycle length = 2^m.	'HC590	SN54LS590
CTR DIVm	Counter with cycle length = m.	'HC160	SN74LS668
RCTRm	Asynchronous (ripple-carry) counter; cycle length = 2^m.	'HC4020	*
ROM	Read-only memory.	*	SN74187
RAM	Random-access read/write memory.	'HC189	SN74170
FIFO	First-in, first-out memory.	*	SN74LS222
I=0	Element powers up cleared to 0 state.	*	SN74AS877
I=1	Element powers up set to 1 state.	'HC7022	SN74AS877
Φ	Highly complex function; "gray box" symbol with limited detail shown under special rules.	*	SN74LS608

*Not all of the general qualifying symbols have been used in TI's CMOS and TTL data books, but they are included here for the sake of completeness.

Table II. Qualifying Symbols for Inputs and Outputs

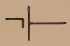
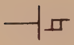
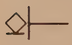
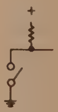
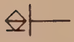
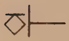
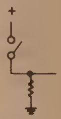
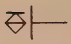
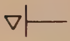
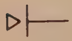
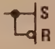
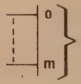
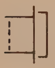
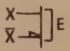
	Logic negation at input. External 0 produces internal 1.
	Logic negation at output. Internal 1 produces external 0.
	Active-low input. Equivalent to in positive logic.
	Active-low output. Equivalent to in positive logic.
	Active-low input in the case of right-to-left signal flow.
	Active-low output in the case of right-to-left signal flow.
	Signal flow from right to left. If not otherwise indicated, signal flow is from left to right.
	Bidirectional signal flow.



	Nonlogic connection. A label inside the symbol will usually define the nature of this pin.
	Input for analog signals (on a digital symbol) (see Figure 14).
	Input for digital signals (on an analog symbol) (see Figure 14).
	Internal connection. 1 state on left produces 1 state on right.
	Negated internal connection. 1 state on left produces 0 state on right.
	Dynamic internal connection. Transition from 0 to 1 on left produces transitory 1 state on right.
	Internal input (virtual input). It always stands at its internal 1 state unless affected by an overriding dependency relationship.
	Internal output (virtual output). Its effect on an internal input to which it is connected is indicated by dependency notation.

The internal (virtual) input is an input originating somewhere else in the circuit and is not connected directly to a terminal. The internal (virtual) output is likewise not connected directly to a terminal. The application of internal inputs and outputs requires an understanding of dependency notation, which is explained in Section 4.

Table III. Symbols Inside the Outline

	Postponed output (of a pulse-triggered flip-flop). The output changes when input initiating change (e.g., a C input) returns to its initial external state or level. See § 5.	
	Bi-threshold input (input with hysteresis).	
	N-P-N open-collector or similar output that can supply a relatively low-impedance L level when not turned off. Requires external pull-up. Capable of positive-logic wired-AND connection.	
	Passive-pull-up output is similar to N-P-N open-collector output but is supplemented with a built-in passive pull-up.	
	N-P-N open-emitter or similar output that can supply a relatively low-impedance H level when not turned off. Requires external pull-down. Capable of positive-logic wired-OR connection.	
	Passive-pull-down output is similar to N-P-N open-emitter output but is supplemented with a built-in passive pull-down.	
	3-state output.	
	Output with more than usual output capability (symbol is oriented in the direction of signal flow).	
EN	Enable input When at its internal 1-state, all outputs are enabled. When at its internal 0-state, open-collector and open-emitter outputs are off, three-state outputs are at normally defined internal logic states and at external high-impedance state, and all other outputs (e.g., totem-poles) are at the internal 0-state.	
J, K, R, S, T	Usual meanings associated with flip-flops (e.g., R = reset, T = toggle)	
D	Data input to a storage element equivalent to: 	
→m ←m	Shift right (left) inputs, m = 1, 2, 3, etc. If m = 1, it is usually not shown.	
+m −m	Counting up (down) inputs, m = 1, 2, 3, etc. If m = 1, it is usually not shown.	
	Binary grouping. m is highest power of 2.	
CT = 15	The contents-setting input, when active, causes the content of a register to take on the indicated value.	
CT = 9	The content output is active if the content of the register is as indicated.	
	Input line grouping . . . indicates two or more terminals used to implement a single logic input. e.g., The paired expander inputs of SN7450. 	
“1”	Fixed-state output always stands at its internal 1 state. For example, see SN74185.	

In an array of elements, if the same general qualifying symbol and the same qualifying symbols associated with inputs and outputs would appear inside each of the elements of the array, these qualifying symbols are usually shown only in the first element. This is done to reduce clutter and to save time in recognition. Similarly, large identical elements that are subdivided into smaller elements may each be represented by an unsubdivided outline. The SN54HC242 or SN54LS440 symbol illustrates this principle.

3.3 Symbols Inside the Outline

Table III shows some symbols used inside the outline. Note particularly that open-collector (open-drain), open-emitter (open-source), and three-state outputs have distinctive symbols. Also note that an EN input affects all of the outputs of the circuit and has no effect on inputs. When an enable input affects only certain outputs and/or affects one or more inputs, a form of dependency notation will indicate this (see 4.9). The effects of the EN input on the various types of outputs are shown.

It is particularly important to note that a D input is always the data input of a storage element. At its internal 1 state, the D input sets the storage element to its 1 state, and at its internal 0 state it resets the storage element to its 0 state.

The binary grouping symbol will be explained more fully in Section 8. Binary-weighted inputs are arranged in order and the binary weights of the least-significant and the most-significant lines are indicated by numbers. In this document weights of input and output lines will be represented by powers of two usually only when the binary grouping symbol is used, otherwise decimal numbers will be used. The grouped inputs generate an internal number on which a mathematical function can be performed or that can be an identifying number for dependency notation (Figure 28). A frequent use is in addresses for memories.

Reversed in direction, the binary grouping symbol can be used with outputs. The concept is analogous to that for the inputs and the weighted outputs will indicate the internal number assumed to be developed within the circuit.

Other symbols are used inside the outlines in accordance with the IEC/IEEE standards but are not shown here. Generally these are associated with arithmetic operations and are self-explanatory.

When nonstandardized information is shown inside an outline, it is usually enclosed in square brackets [like these].

4.0 DEPENDENCY NOTATION

4.1 General Explanation

Dependency notation is the powerful tool that sets the IEC symbols apart from previous systems and makes compact, meaningful, symbols possible. It provides the means of denoting the relationship between inputs, outputs, or inputs and outputs without actually showing all the elements and interconnections involved. The information provided by dependency notation supplements that provided by the qualifying symbols for an element's function.

In the convention for the dependency notation, use will be made of the terms ''affecting'' and ''affected.'' In cases where it is not evident which inputs must be considered as being the affecting or the affected ones (e.g., if they stand in an AND relationship), the choice may be made in any convenient way.

So far, eleven types of dependency have been defined and all of these are used in various TI data books. X dependency is used mainly with CMOS circuits. They are listed below in the order in which they are presented and are summarized in Table IV following 4.12.

Section	Dependency Type or Other Subject
4.2	G, AND
4.3	General Rules for Dependency Notation
4.4	V, OR
4.5	N, Negate (Exclusive-OR)
4.6	Z, Interconnection
4.7	X, Transmission
4.8	C, Control
4.9	S, Set and R, Reset
4.10	EN, Enable
4.11	M, Mode
4.12	A, Address

4.2 G (AND) Dependency

A common relationship between two signals is to have them ANDed together. This has traditionally been shown by explicitly drawing an AND gate with the signals connected to the inputs of the gate. The 1972 IEC publication and the 1973 IEEE/ANSI standard showed several ways to show this AND relationship using dependency notation. While ten other forms of dependency have since been defined, the ways to invoke AND dependency are now reduced to one.

In Figure 4 input **b** is ANDed with input **a** and the complement of **b** is ANDed with **c**. The letter G has been chosen to indicate AND relationships and is placed at input **b**, inside the symbol. A number considered appropriate by the symbol designer (1 has been used here) is placed after the letter G and also at each affected input. Note the bar over the 1 at input **c**.

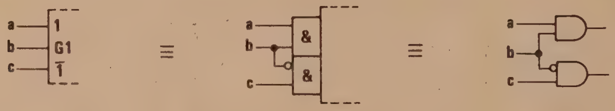


Figure 4. G Dependency Between Inputs

In Figure 5, output **b** affects input **a** with an AND relationship. The lower example shows that it is the internal logic state of **b**, unaffected by the negation sign, that is ANDed. Figure 6 shows input **a** to be ANDed with a dynamic input **b**.

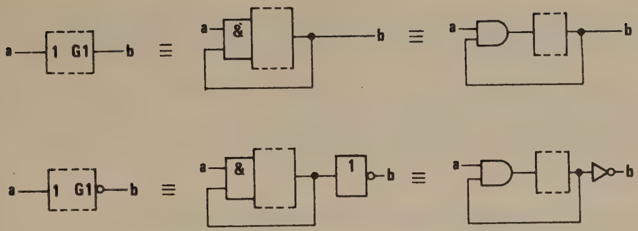


Figure 5. G Dependency Between Outputs and INputs

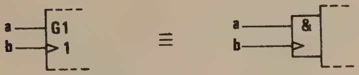


Figure 6. G Dependency with a Dynamic Input

The rules for G dependency can be summarized thus:

> When a G*m* input or output (*m* is a number) stands at its internal 1 state, all inputs and outputs affected by G*m* stand at their normally defined internal logic states. When the G*m* input or output stands at its 0 state, all inputs and outputs affected by G*m* stand at their internal 0 states.

4.3 Conventions for the Application of Dependency Notation in General

The rules for applying dependency relationships in general follow the same pattern as was illustrated for G dependency.

Application of dependency notation is accomplished by:

1) labeling the input (or output) *affecting* other inputs or outputs with the letter symbol indicating the relationship involved (e.g., G for AND) followed by an identifying number, appropriately chosen, and
2) labeling each input or output *affected* by that affecting input (or output) with that same number.

If it is the complement of the internal logic state of the affecting input or output that does the affecting, then a bar is placed over the identifying numbers at the affected inputs or outputs (Figure 4).

If two affecting inputs or outputs have the same letter and same identifying number, they stand in an OR relationship to each other (Figure 7).

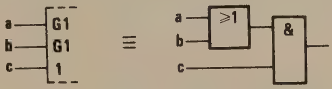


Figure 7. ORed Affecting Inputs

If the affected input or output requires a label to denote its function (e.g., ''D''), this label will be *prefixed* by the identifying number of the affecting input (Figure 15).

If an input or output is affected by more than one affecting input, the identifying numbers of each of the affecting inputs will appear in the label of the affected one, separated by commas. The normal reading order of these numbers is the same as the sequence of the affecting relationships (Figure 15).

If the labels denoting the functions of affected inputs or outputs must be numbers (e.g., outputs of a coder), the identifying numbers to be associated with both affecting inputs and affected inputs or outputs will be replaced by another character selected to avoid ambiguity, e.g., Greek letters (Figure 8).

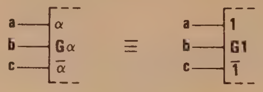


Figure 8. Substitution for Numbers

4.4 V (OR) Dependency

The symbol denoting OR dependency is the letter V (Figure 9).

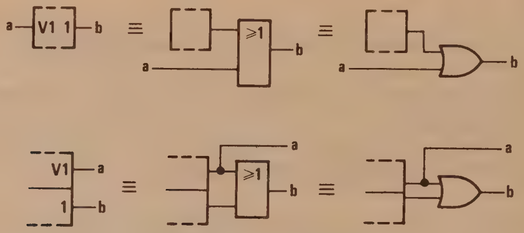
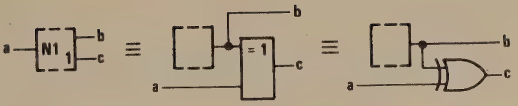


Figure 9. V (OR) Dependency

When a Vm input or output stands at its internal 1 state, all inputs and outputs affected by Vm stand at their internal 1 states. When the Vm input or output stands at its internal 0 state, all inputs and outputs affected by Vm stand at their normally defined internal logic states.

4.5 N (Negate) (Exclusive-OR) Dependency

The symbol denoting negate dependency is the letter N (Figure 10). Each input or output affected by an Nm input or output stands in an Exclusive-OR relationship with the Nm input or output.



If a = 0, then c = b
If a = 1, then c = b

Figure 10. N (Negate) (Exclusive-OR) Dependency

When an N*m* input or output stands at its internal 1 state, the internal logic state of each input and each output affected by N*m* is the complement of what it would otherwise be. When an N*m* input or output stands at its internal 0 state, all inputs and outputs affected by N*m* stand at their normally defined internal logic states.

4.6 Z (Interconnection) Dependency

The symbol denoting interconnection dependency is the letter Z.

Interconnection dependency is used to indicate the existence of internal logic connections between inputs, outputs, internal inputs, and/or internal outputs.

The internal logic state of an input or output affected by a Z*m* input or output will be the same as the internal logic state of the Z*m* input or output, unless modified by additional dependency notation (Figure 11).

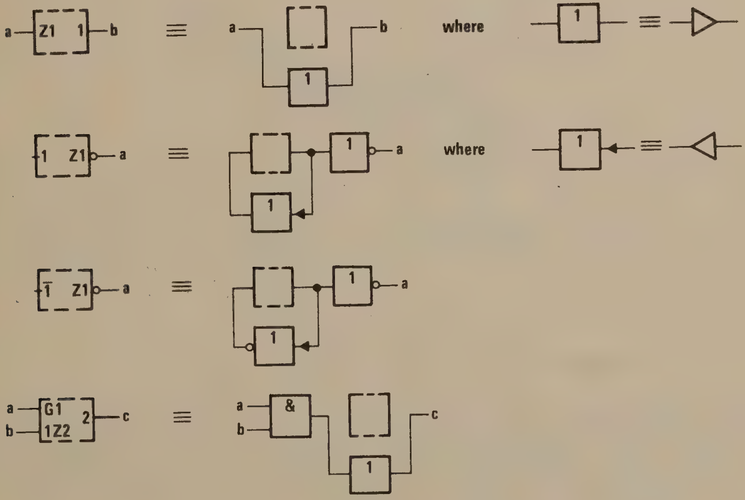
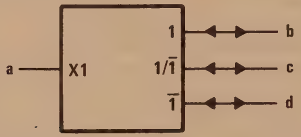


Figure 11. Z (Interconnection) Dependency

4.7 X (Transmission) Dependency

The symbol denoting transmission dependency is the letter X.

Transmission dependency is used to indicate controlled bidirectional connections between affected input/output ports (Figure 12).



If a = 1, there is a bidirectional connection between b and c.

If a = 0, there is a bidirectional connection between c and d.

Figure 12. X (Transmission) Dependency

When an X*m* input or output stands at its internal 1 state, all input-output ports affected by this X*m* input or output are bidirectionally connected together and stand at the same internal logic state or analog signal level. When an X*m* input or output stands at its internal 0 state, the connection associated with this set of dependency notation does not exist.

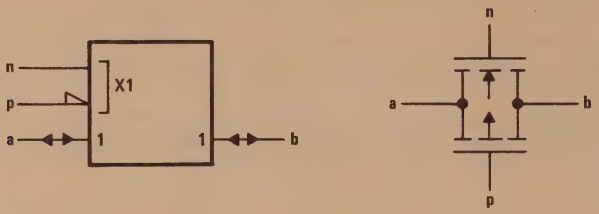


Figure 13. CMOS Transmission Gate Symbol and Schematic

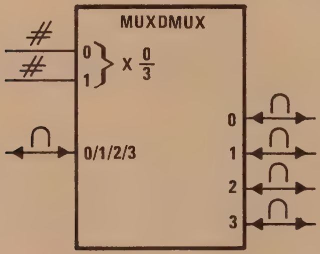


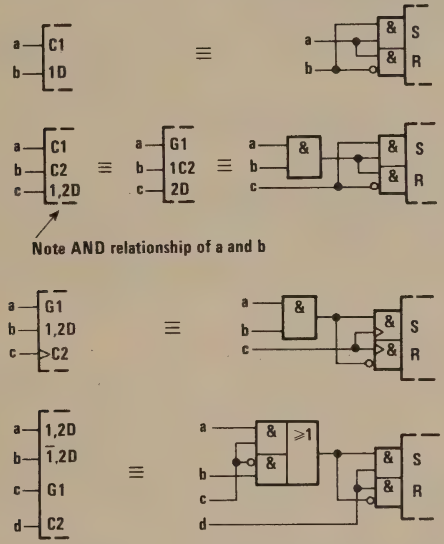
Figure 14. Analog Data Selector (Multiplexer/Demultiplexer)

Although the transmission paths represented by X dependency are inherently bidirectinal, use is not always made of this property. This is analogous to a piece of wire, which may be constrained to carry current in only one direction. If this is the case in a particular application, then the directional arrows shown in Figures 12, 13, and 14 would be omitted.

4.8 C (Control) Dependency

The symbol denoting control dependency is the letter C.

Control inputs are usually used to enable or disable the data (D, J, K, R, or S) inputs of storage elements. They may take on their internal 1 states (be active) either statically or dynamically. In the latter case the dynamic input symbol is used as shown in the third example of Figure 15.



Note AND relationship of a and b

Input c selects which of a or b is stored when d goes low.

Figure 15. C (Control) Dependency

When a C*m* input or output stands at its internal 1 state, the inputs affected by C*m* have their normally defined effect on the function of the element, i.e., these inputs are enabled. When a C*m* input or output stands at its internal 0 state, the inputs affected by C*m* are disabled and have no effect on the function of the element.

4.9 S (Set) and R (Reset) Dependencies

The symbol denoting set dependency is the letter S. The symbol denoting reset dependency is the letter R.

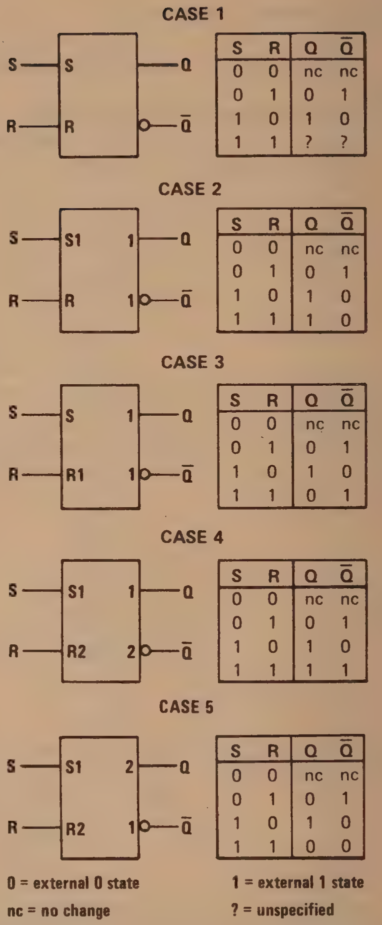
Set and reset dependencies are used if it is necessary to specify the effect of the combination R = S = 1 on a bistable element. Case 1 in Figure 16 does not use S or R dependency.

When an S*m* input is at its internal 1 state, outputs affected by the S*m* input will react, regardless of the state of an R input, as they normally would react to the combination S = 1, R = 0. See cases 2, 4, and 5 in Figure 16.

When an R*m* input is at its internal 1 state, outputs affected by the R*m* input will react, regardless of the state of an S input, as they normally would react to the combination S = 0, R = 1. See cases 3, 4, and 5 in Figure 16.

When an S*m* or R*m* input is at its internal 0 state, it has no effect.

Note that the noncomplementary output patterns in cases 4 and 5 are only pseudo stable. The simultaneous return of the inputs to S = R = 0 produces an unforeseeable stable and complementary output pattern.



CASE 1

S	R	Q	$\bar{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	?	?

CASE 2

S	R	Q	$\bar{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	1	0

CASE 3

S	R	Q	$\bar{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	0	1

CASE 4

S	R	Q	$\bar{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	1	1

CASE 5

S	R	Q	$\bar{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	0	0

0 = external 0 state
1 = external 1 state
nc = no change
? = unspecified

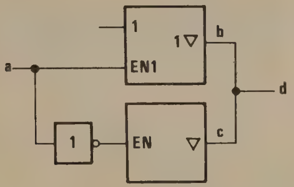
Figure 16. S (Set) and R (Reset) Dependencies

4.10 EN (Enable) Dependency

The symbol denoting enable dependency is the combination of letters EN.

An EN*m* input has the same effect on outputs as an EN input, see 3.1, but it affects only those outputs labeled with the identifying number *m*. It also affects those inputs labeled with the identifying number *m*. By contrast, an EN input affects all outputs and no inputs. The effect of an EN*m* input on an affected input is identical to that of a C*m* input (Figure 17).

When an EN*m* input stands at its internal 1 state, the inputs affected by EN*m* have their normally defined effect on the function of the element and the outputs affected by this input stand at their normally defined internal logic states, i.e., these inputs and outputs are enabled.



If a = 0, b is disabled and d = c
If a = 1, c is disabled and d = b

Figure 17. EN (Enable) Dependency

When an EN*m* input stands at its internal 0 state, the inputs affected by EN*m* are disabled and have no effect on the function of the element, and the outputs affected by EN*m* are also disabled. Open-collector outputs are turned off, three-state outputs stand at their normally defined internal logic states but externally exhibit high impedance, and all other outputs (e.g., totem-pole outputs) stand at their internal 0 states.

4.11 M (MODE) Dependency

The symbol denoting mode dependency is the letter M.

Mode dependency is used to indicate that the effects of particular inputs and outputs of an element depend on the mode in which the element is operating.

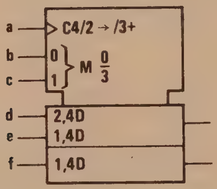
If an input or output has the same effect in different modes of operation, the identifying numbers of the relevant affecting M*m* inputs will appear in the label of that affected input or output between parentheses and separated by solidi (Figure 22).

4.11.1 M Dependency Affecting Inputs

M dependency affects inputs the same as C dependency. When an M*m* input or M*m* output stands at its internal 1 state, the inputs affected by this M*m* input or M*m* output have their normally defined effect on the function of the element, i.e., the inputs are enabled.

When an M*m* input or M*m* output stands at its internal 0 state, the inputs affected by this M*m* input or M*m* output have no effect on the function of the element. When an affected input has several sets of labels separated by solidi (e.g., C4/2→/3+), any set in which the identifying number of the M*m* input or M*m* output appears has no effect and is to be ignored. This represents disabling of some of the functions of a multifunction input.

The circuit in Figure 18 has two inputs, **b** and **c**, that control which one of four modes (0, 1, 2, or 3) will exist at any time. Inputs **d**, **e**, and **f** are D inputs subject to dynamic control (clocking) by the **a** input. The numbers 1 and 2 are in the series chosen to indicate the modes so inputs **e** and **f** are only enabled in mode 1 (for parallel loading) and input **d** is only enabled in mode 2 (for serial loading). Note that input **a** has three functions. It is the clock for entering data. In mode 2, it causes right shifting of data, which means a shift away from the control block. In mode 3, it causes the contents of the register to be incremented by one count.



Note that all operations are synchronous.

In MODE 0 (b = 0, c = 0), the outputs remain at their existing states as none of the inputs has an effect.

In MODE 1 (b = 1, c = 0), parallel loading takes place thru inputs e and f.

In MODE 2 (b = 0, c = 1), shifting down and serial loading thru input d take place.

In MODE 3 (b = c = 1), counting up by increment of 1 per clock pulse takes place.

Figure 18. M (Mode) Dependency Affecting Inputs

4.11.2 M Dependency Affecting Outputs

When an M*m* input or M*m* output stands at its internal 1 state, the affected outputs stand at their normally defined internal logic states, i.e., the outputs are enabled.

When an M*m* input or M*m* output stands at its internal 0 state, at each affected output any set of labels containing the identifying number of that M*m* input or M*m* output has no effect and is to be ignored. When an output has several different sets of labels separated by solidi (e.g., 2,4/3,5), only those sets in which the identifying number of this M*m* input or M*m* output appears are to be ignored.

Figure 19 shows a symbol for a device whose output can behave like either a 3-state output or an open-collector output depending on the signal applied to input **a**. Mode 1 exists when input **a** stands at its internal 1 state and, in that case, the three-state symbol applies and the open-element symbol has no effect. When **a** = 0, mode 1 does not exist so the three-state symbol has no effect and the open-element symbol applies.

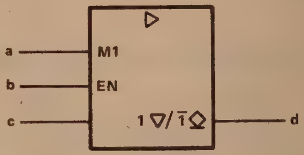


Figure 19. Type of Output Determined by Mode

In Figure 20, if input **a** stands at its internal 1 state establishing mode 1, output **b** will stand at its internal 1 state only when the content of the register equals 9. Since output **b** is located in the common-control block with no defined function outside of mode 1, the state of this output outside of mode 1 is not defined by the symbol.

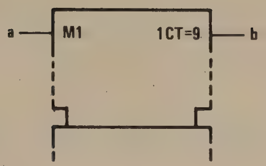


Figure 20. An Output of the Common-Control Block

In Figure 21, if input **a** stands at its internal 1 state establishing mode 1, output **b** will stand at its internal 1 state only when the content of the register equals 15. If input **a** stands at its internal 0 state, output **b** will stand at its internal 1 state only when the content of the register equals 0.

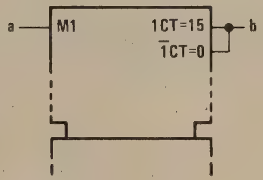


Figure 21. Determining and Output's Function

In Figure 22 inputs **a** and **b** are binary weighted to generate the numbers 0, 1, 2, or 3. This determines which one of the four modes exists.

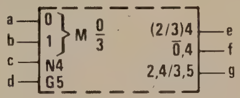


Figure 22. Dependent Relationships Affected by Mode

At output **e** the label set causing negation (if **c** = 1) is effective only in modes 2 and 3. In modes 0 and 1 this output stands at its normally defined state as if it had no labels. At output **f** the label set has effect when the mode is not 0 so output **e** is negated (if **c** = 1) in modes 1, 2, and 3. In mode 0 the label set has no effect so the output stands at its normally defined state. In this example $\overline{0}$,4 is equivalent to (1/2/3)4. At output **g** there are two label sets. The first set, causing negation (if **c** = 1), is effective only in mode 2. The second set, subjecting **g** to AND dependency on **d**, has effect only in mode 3.

Note that in mode 0 none of the dependency relationships has any effect on the outputs, so **e**, **f**, and **g** will all stand at the same state.

4.12 A (Address) Dependency

The symbol denoting address dependency is the letter A.

Address dependency provides a clear representation of those elements, particularly memories, that use address control inputs to select specified sections of a multildimensional arrays. Such a section of a memory array is usually called a word. The purpose of address dependency is to allow a symbolic presentation of the entire array. An input of the array shown at a particular element of this general section is common to the corresponding elements of all selected sections of the array. An output of the array shown at a particular element of this general section is the result of the OR function of the outputs of the corresponding elements of selected sections.

Inputs that are not affected by any affecting address input have their normally defined effect on all sections of the array, whereas inputs affected by an address input have their normally defined effect only on the section selected by that address input.

An affecting address input is labeled with the letter A followed by an identifying number that corresponds with the address of the particular section of the array selected by this input. Within the general section presented by the symbol, inputs and outputs affected by an A*m* input are labeled with the letter A, which stands for the identifying numbers, i.e., the addresses, of the particular sections.

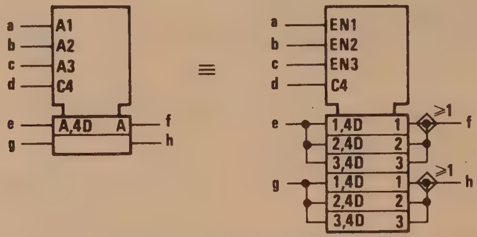


Figure 23. A (Address) Dependency

Figure 23 shows a 3-word by 2-bit memory having a separate address line for each word and uses EN dependency to explain the operation. To select word 1, input **a** is taken to its 1 state, which establishes mode 1. Data can now be clocked into the inputs marked "1,4D." Unless words 2 and 3 are also selected, data cannot be clocked in at the inputs marked "2,4D" and "3,4D." The outputs will be the OR functions of the selected outputs, i.e., only those enabled by the active EN functions.

The identifying numbers of affecting address inputs correspond with the addresses of the sections selected by these inputs. They need not necessarily differ from those of other affecting dependency-inputs (e.g., G, V, N, . . .), because in the general section presented by the symbol they are replaced by the letter A.

If there are several sets of affecting A*m* inputs for the purpose of independent and possibly simultaneous access to sections of the array, then the letter A is modified to 1A, 2A, ... Because they have access to the same sections of the array, these sets of A inputs may have the same identifying numbers. The symbols for 'HC170 or SN74LS170 make use of this.

Figure 24 is another illustration of the concept.

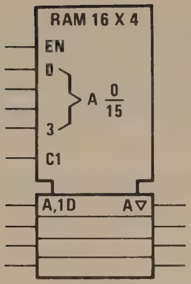


Figure 24. Array of 16 Sections of Four Transparent Latches with 3-State Outputs Comprising a 16-Word X 4-Bit Random-Access Memory

Table IV. Summary of Dependency Notation

TYPE OF DEPENDENCY	LETTER SYMBOL*	AFFECTING INPUT AT ITS 1-STATE	AFFECTING INPUT AT ITS 0-STATE
Address	A	Permits action (address selected)	Prevents action (address not selected)
Control	C	Permits action	Prevents action
Enable	EN	Permits action	Prevents action of inputs ◇outputs off ▽outputs at external high impedance, no change in internal logic state Other outputs at internal 0 state
AND	G	Permits action	Imposes 0 state
Mode	M	Permits action (mode selected)	Prevents action (mode not selected)
Negate (Ex-OR)	N	Complements state	No effect
Reset	R	Affected output reacts as it would to S = 0, R = 1	No effect
Set	S	Affected output reacts as it would to S = 1, R = 0	No effect
OR	V	Imposes 1 state	Permits action
Transmission	X	Bidirectional connection exists	Bidirectional connection does not exist
Interconnection	Z	Imposes 1 state	Imposes 0 state

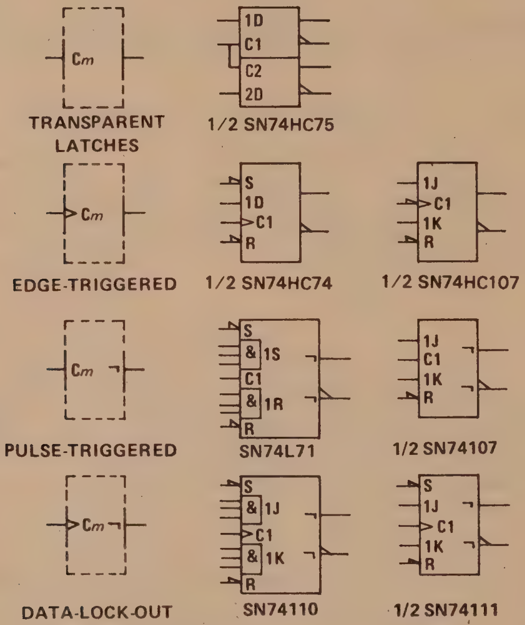
*These letter symbols appear at the AFFECTING input (or output) and are followed by a number. Each input (or output) AFFECTED by that input is labeled with that same number. When the labels EN, R, and S appear at inputs without the following numbers, the descriptions above do not apply. The action of these inputs is described under "Symbols Inside the Outline," see 3.3.

5.0 BISTABLE ELEMENTS

The dynamic input symbol, the postponed output symbol, and dependency notation provide the tools to differentiate four main types of bistable elements and make synchronous and asynchronous inputs easily recognizable (Figure 25). The first column shows the essential distinguishing features; the other columns show examples.

Transparent latches have a level-operated control input. The D input is active as long as the C input is at its internal 1 state. The outputs respond immediately. Edge-triggered elements accept data from D, J, K, R, or S inputs on the active transition of C. Pulse-triggered elements require the setup of data before the start of the control pulse; the C input is considered static since the data must be maintained as long as C is at its 1 state. The output is postponed until C returns to its 0 state. The data-lock-out element is similar to the pulse-triggered version except that the C input is considered dynamic in that shortly after C goes through its active transition, the data inputs are disabled and data does not have to be held. However, the output is still postponed until the C input returns to its initial external level.

Notice that synchronous inputs can be readily recognized by their dependency labels (1D, 1J, 1K, 1S, 1R) compared to the asynchronous inputs (S, R), which are not dependent on the C inputs.



TRANSPARENT LATCHES

1/2 SN74HC75

EDGE-TRIGGERED

1/2 SN74HC74

1/2 SN74HC107

PULSE-TRIGGERED

SN74L71

1/2 SN74107

DATA-LOCK-OUT

SN74110

1/2 SN74111

Figure 25. Four Types of Bistable Circuits

6.0 CODERS

The general symbol for a coder or code converter is shown in Figure 26. X and Y may be replaced by appropriate indications of the code used to represent the information at the inputs and at the outputs, respectively.

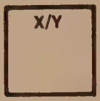


Figure 26. Coder General Symbol

Indication of code conversion is based on the following rule:

> Depending on the input code, the internal logic states of the inputs determine an internal value. This value is reproduced by the internal logic states of the outputs, depending on the output code.

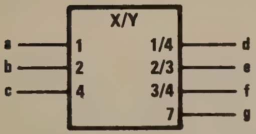
The indication of the relationships between the internal logic states of the inputs and the internal value is accomplished by:

1) labeling the inputs with numbers. In this case the internal value equals the sum of the weights associated with those inputs that stand at their internal 1-state, or by
2) replacing X by an appropriate indication of the input code and labeling the inputs with characters that refer to this code.

The relationships between the internal value and the internal logic states of the outputs are indicated by:

1) labeling each output with a list of numbers representing those internal values that lead to the internal 1-state of that output. These numbers shall be separated by solidi as in Figure 27. This labeling may also be applied when Y is replaced by a letter denoting a type of dependency (see Section 7). If a continuous range of internal values produces the internal 1 state of an output, this can be indicated by two numbers that are inclusively the beginning and the end of the range, with these two numbers separated by three dots (e.g., 4 . . . 9 = 4/5/6/7/8/9) or by
2) replacing Y by an appropriate indiction of the output code and labeling the outputs with characters that refer to this code as in Figure 28.

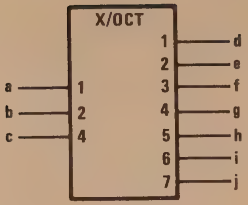
Alternatively, the general symbol may be used together with an appropriate reference to a table in which the relationship between the inputs and outputs is indicated. This is a recommended way to symbolize a PROM after it has been programmed.



FUNCTION TABLE

INPUTS			OUTPUTS			
c	b	a	g	f	e	d
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

Figure 27. An X/Y Code Converter



FUNCTION TABLE

INPUTS			OUTPUTS						
c	b	a	j	i	h	g	f	e	d
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Figure 28. An X/Octal Code Converter

7.0 USE OF A CODER TO PRODUCE AFFECTING INPUTS

It often occurs that a set of affecting inputs for dependency notation is produced by decoding the signals on certain inputs to an element. In such a case use can be made of the symbol for a coder as an embedded symbol (Figure 29).

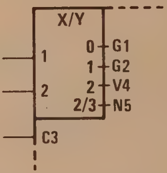


Figure 29. Producing Various Types of Dependencies

If all affecting inputs produced by a coder are of the same type and their identifying numbers shown at the outputs of the coder, Y (in the qualifying symbol X/Y) may be replaced by the letter denoting the type of dependency. The indications of the affecting inputs should then be omitted (Figure 30).

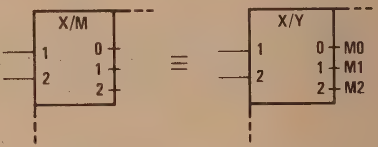


Figure 30. Producing One Type of Dependency

8.0 USE OF BINARY GROUPING TO PRODUCE AFFECTING INPUTS

If all affecting inputs produced by a coder are of the same type and have consecutive identifying numbers not necessarily corresponding with the numbers that would have been shown at the outputs of the coder, use can be made of the binary grouping symbol. k external lines effectively generate 2^k internal inputs. The bracket is followed by the letter denoting the type of dependency followed by m1/m2. The m1 is to be replaced by the smallest identifying number and the m2 by the largest one, as shown in Figure 31.

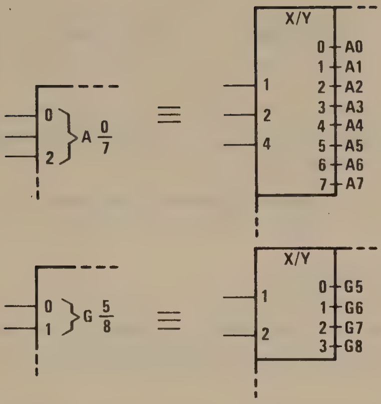


Figure 31. Use of the Binary Grouping Symbol

9.0 SEQUENCE OF INPUT LABELS

If an input having a single functional effect is affected by other inputs, the qualifying symbol (if there is any) for that functional effect is preceded by the labels corresponding to the affecting inputs. The left-to-right order of these preceding labels is the order in which the effects or modifications must be applied. The affected input has no functional effect on the element if the logic state of any one of the affecting inputs, considered separately, would cause the affected input to have no effect, regardless of the logic states of other affecting inputs.

If an input has several different functional effects or has several different sets of affecting inputs, depending on the mode of action, the input may be shown as often as required. However, there are cases in which this method of presentation is not advantageous. In those cases the input may be shown once with the different sets of labels separated by solidi (Figure 32). No meaning is attached to the order of these sets of labels. If one of the functional effects of an input is that of an unlabeled input to the element, a solidus will precede the first set of labels shown.

If all inputs of a combinational element are disabled (caused to have no effect on the function of the element), the internal logic states of the outputs of the element are not specified by the symbol. If all inputs of a sequential element are disabled, the content of this element is not changed and the outputs remain at their existing internal logic states.

Labels may be factored using algebraic techniques (Figure 33).

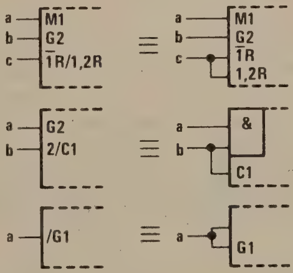


Figure 32. Input Labels

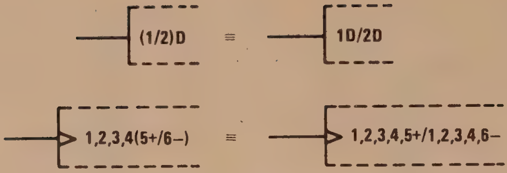


Figure 33. Factoring Input Labels

10.0 SEQUENCE OF OUTPUT LABELS

If an output has a number of different labels, regardless of whether they are identifying numbers of affecting inputs or outputs or not, these labels are shown in the following order:

1) If the postponed output symbol has to be shown, this comes first, if necessary preceded by the indications of the inputs to which it must be applied
2) Followed by the labels indicating modifications of the internal logic state of the output, such that the left-to-right order of these labels corresponds with the order in which their effects must be applied
3) Followed by the label indicating the effect of the output on inputs and other outputs of the element.

Symbols for open-circuit or three-state outputs, where applicable, are placed just inside the outside boundary of the symbol adjacent to the output line (Figure 34).

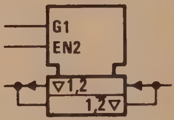


Figure 34. Placement of 3-State Symbols

If an output needs several different sets of labels that represent alternative functions (e.g., depending on the mode of action), these sets may be shown on different output lines that must be connected outside the outline. However, there are cases in which this method of presentation is not advantageous. In those cases the output may be shown once with the different sets of labels separated by solidi (Figure 35).

Two adjacent identifying numbers of affecting inputs in a set of labels that are not already separated by a nonnumeric character should be separated by a comma.

If a set of labels of an output not containing a solidus contains the identifying number of an affecting M*m* input standing at its internal 0 state, this set of labels has no effect on that output.

Labels may be factored using algebraic techniques (Figure 36).

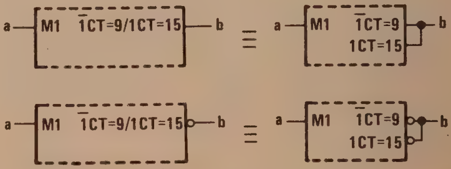


Figure 35. Output Labels

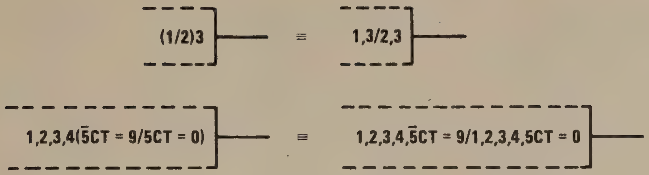


Figure 36. Factoring Output Labels

F.A. Mann received his Bachelor of Science degree from the United States Naval Academy, Annapolis, MD in 1953 and his Master's degree in Engineering Administration from Southern Methodist University, Dallas, TX in 1970. He joined Texas Instruments in 1957 and has worked as a Manufacturing Engineer, Product Marketing Engineer, and Semiconductor Data Sheet Manager. Currently, he serves as Technical Advisor for data sheets and other technical documentation for semiconductors. He has served on many JEDEC committees since 1963 and has been a USA delegate to the technical committee for semiconductor devices of the International Electrotechnical Commission since 1970. He is a member of the IEEE committees for Logic Symbols and for Logic Diagrams, and participates in meetings of an international working group on logic symbols.

Mr. Mann can be reached at (214) 995-2867 in Dallas.

Texas Instruments Semiconductor Technical Literature

TTL Data Book, Vol. 1, 1984, 336 pages.
Product guide for all TI TTL devices, functional indexes, alphanumeric index, and general information.

TTL Data Book, Vol. 2, 1984, 1,000 pages.
Detailed specifications and application information on the TI family of Low-power Schottky (LS), Schottky (S), and standard TTL logic devices.

TTL Data Book, Vol. 3, 1984, 792 pages.
Detailed specifications and application information on the TI family of Advanced Low-power Schottky (ALS) and Advanced Schottky (AS) logic devices.

TTL Data Book, Vol. 4, 1984, 416 pages.
Detailed specifications and application information on the TI family of bipolar field-programmable logic (FPL), programmable read-only memories (PROM), random-access memories (RAM), microprocessors, and support circuits.

High-speed CMOS Logic Data Book, 1984, 580 pages.
Detailed specifications and application information on the TI family of High-speed CMOS logic devices. Includes product selection guide, glossary, and alphanumeric index.

Linear Circuits Data Book, 1984, 820 pages.
Detailed specifications on operational amplifiers, voltage comparators, voltage regulators, data-acquisition devices, a/d converters, timers, switches, amplifiers, and special functions. Includes LinCMOS™ functions. Contains product guide, interchangeability guide, glossary, and alphanumeric index.

Interface Circuits Data Book, 1981, 700 pages.
Includes specifications and applications information on TTL logic interface circuits, as well as product profiles on the line drivers/receivers and peripheral drivers.

Optoelectronics Data Book, 1983, 480 pages.
Contains more than 300 device types representing traditional optoelectronics (IREDs, LEDs, detectors, couplers, and displays), special components (avalanche, photodiodes, and transimpedance amplifiers), fiber optic components (sources, detectors, and interconnecting cables), and new image sensors (linear and arrays).

MOS Memory Data Book, 1984, 456 pages.
Detailed specifications on dynamic RAMs, static RAMs, EPROMs, ROMs, cache address comparators, and memory controllers. Contains product guide, interchangeability guide, glossary, and alphanumeric index. Also, chapters on testing and reliability.

TMS7000 Family Data Manual, 1983, 350 pages.
Detailed specifications and application information on TI's family of microprogrammable 8-bit microcomputers. Includes architecture description, device operation, instruction set, electrical characteristics, and mechanical data. TMS7000 microcomputers include versions in CMOS and SMOS and with on-board UART.

TMSxxxxx Microcomputer Data Manuals
These manuals contain detailed specifications and application information on specific TMSxxxxx microcomputers and peripherals. Include architecture description, device operation, instruction set, electrical characteristics, and mechanical data.

Assembly Language Programmer's Guides.
TMS32010, 1983, 160 pages.
TMS99000, 1983, 322 pages.
TMS7000, 1983, 160 pages.
Include general programming information, assembly instructions, assembler directives, assembler output, and application notes.

TMS32010 User's Guide, 1984, 400 pages.
Detailed application information on the TMS32010 Digital Signal Processor. Detailed reference manual on use of the TMS320 instruction set. Data sheets included.

Fundamentals of Microcomputer Design, 1982, 584 pages.
University textbook. Subjects include microprocessors, software, instruction sets, microcomputer programming, high-level languages, hardware features, microcomputer memory, and I/O design. A design example is included.

Understanding Series™ Books
The Understanding Series books form a library written for anyone who wants to learn quickly and easily about today's technology, its impact on our world, and its application in our lives. Each book is written in bright, clear, down-to-earth language and focuses on one aspect of what's new in today's electronics. Engineering concepts and theory are explained using simple arithmetic. Technical terms are explained in layman's language. Ideal for self-paced, individualized instruction. Currently 13 different titles in the series.

Write for current availability and prices to:
Texas Instruments
Information Publishing Center
P.O. Box 225012 MS-54
Dallas, TX 75265

April 1984

TI Sales Offices

ALABAMA: Huntsville, (205) 837-7530.

ARIZONA: Phoenix, (602) 995-1007.

CALIFORNIA: Irvine, (714) 660-1200; **Sacramento,** (916) 929-1521; **San Diego,** (714) 278-9600; **Santa Clara,** (408) 980-9000; **Torrance,** (213) 973-2571; **Woodland Hills,** (213) 704-7759.

COLORADO: Aurora, (303) 695-2800.

CONNECTICUT: Wallingford, (203) 269-0074.

FLORIDA: Ft. Lauderdale, (305) 973-8502; **Maitland,** (305) 660-4600; **Tampa,** (813) 870-6420.

GEORGIA: Atlanta, (404) 452-4600.

ILLINOIS: Arlington Heights, (312) 640-3000.

INDIANA: Ft. Wayne, (219) 424-5174; **Indianapolis,** (317) 248-8555.

IOWA: Cedar Rapids, (319) 395-9550.

MARYLAND: Baltimore, (301) 944-8600.

MASSACHUSETTS: Waltham, (617) 895-9100.

MICHIGAN: Farmington Hills, (313) 553-1500.

MINNESOTA: Edina, (612) 830-1600.

MISSOURI: Kansas City, (816) 523-2500; **St. Louis,** (314) 569-7600.

NEW JERSEY: Clark, (201) 574-9800.

NEW MEXICO: Albuquerque, (505) 345-2555.

NEW YORK: East Syracuse, (315) 463-9291; **Endicott,** (607) 754-3900; **Melville,** (516) 454-6600; **Poughkeepsie,** (914) 473-2900; **Rochester,** (716) 424-5400.

NORTH CAROLINA: Charlotte, (704) 527-0930; **Raleigh,** (919) 876-2725.

OHIO: Beachwood, (216) 464-6100; **Dayton,** (513) 258-3877.

OKLAHOMA: Tulsa, (918) 250-0633.

OREGON: Beaverton, (503) 643-6758.

PENNSYLVANIA: Ft. Washington, (215) 643-6450; **Coraopolis,** (412) 771-8550.

TEXAS: Austin, (512) 250-7655; **Houston,** (713) 778-6592; **Richardson,** (214) 680-5082; **San Antonio,** (512) 496-1779.

UTAH: Murray, (801) 266-8972.

VIRGINIA: Fairfax, (703) 849-1400.

WISCONSIN: Brookfield, (414) 785-7140.

WASHINGTON: Redmond, (206) 881-3080.

CANADA: Nepean, Ontario (613) 726-1970; **Richmond Hill, Ontario** (416) 884-9181; **St. Laurent, Quebec** (514) 334-3635. F

TI Regional Technology Centers

CALIFORNIA: Irvine, (714) 660-8140, Hotline: (714) 660-8164. **Santa Clara** (408) 748-2220, Hotline: (408) 980-0305.

GEORGIA: Atlanta, (404) 452-4682, Hotline: (404) 452-4686.

ILLINOIS: Chicago, (312) 640-2909, Hotline: (312) 228-6008.

MASSACHUSETTS: Boston, (617) 890-6671, Hotline: (617) 890-4271.

TEXAS: Dallas, (214) 680-5066, Hotline: (214) 680-5096.

TI Distributors

TI AUTHORIZED DISTRIBUTORS IN U.S.:

- Arrow Electronics
- Diplomat, Inc.
- ESCO Electronics
- General Radio Supply Company
- Graham Electronics
- Hall-Mark Electronics
- Harrison Equipment Co.
- International Electronics
- JACO Electronics
- Kierulff Electronics
- LCOMP, Incorporated
- Marshall Industries
- Milgray Electronics
- Newark Electronics
- Rochester Radio Supply
- Time Electronics
- R. V. Weatherford Co.
- Wyle Laboratories

TI AUTHORIZED DISTRIBUTORS IN CANADA:

- CESCO Electronics, Inc.
- Future Electronics
- ITT Components
- L. A. Varah, Ltd.

ALABAMA: Arrow (205) 882-2730; Marshall (205) 881-9235.

ARIZONA: Phoenix, Arrow (602) 968-4800; Kierulff (602) 243-4101; Marshall (602) 968-6181; Wyle (602) 249-2232; **Tucson,** Kierulff (602) 624-9986.

CALIFORNIA: Los Angeles/Orange County, Arrow (213) 701-7500, (714) 838-5422; Kierulff (213) 725-0325, (714) 731-5711; Marshall (213) 999-5001, (213) 442-7204, (714) 556-6400; R.V. Weatherford (714) 634-9600, (213) 849-3451, (714) 623-1261; Wyle (213) 322-8100, (714) 863-9953; **Sacramento,** Arrow (916) 925-7456; Wyle (916) 638-5282; **San Diego,** Arrow (619) 565-4800; Kierulff (619) 278-2112; Marshall (619) 578-9600; Wyle (619) 565-9171; **San Francisco Bay Area,** Arrow (408) 745-6600; Kierulff (415) 968-6292; Marshall (408) 732-1100; Wyle (408) 727-2500; **Santa Barbara,** R. V. Weatherford (805) 965-8551.

COLORADO: Arrow (303) 696-1111; Kierulff (303) 790-4444; Wyle (303) 457-9953.

CONNECTICUT: Arrow (203) 265-7741; Diplomat (203) 797-9674; Kierulff (203) 265-1115; Marshall (203) 265-3822; Milgray (203) 795-0714.

FLORIDA: Ft. Lauderdale, Arrow (305) 776-7790; Diplomat (305) 974-8700; Kierulff (305) 486-4004; **Orlando,** Arrow (305) 725-1480; Milgray (305) 647-5747; **Tampa,** Diplomat (813) 443-4514; Kierulff (813) 576-1966.

TEXAS
INSTRUMENTS

Creating useful products
and services for you.

GEORGIA: Arrow (404) 449-8252; Kierulff (404) 447-5252; Marshall (404) 923-5750.

ILLINOIS: Arrow (312) 397-3440; Diplomat (312) 595-1000; Hall-Mark (312) 860-3800; Kierulff (312) 640-0200; Newark (312) 638-4411.

INDIANA: Indianapolis, Arrow (317) 243-9353; Graham (317) 634-8202; **Ft. Wayne,** Graham (219) 423-3422.

IOWA: Arrow (319) 395-7230.

KANSAS: Kansas City, Hall-Mark (913) 888-4747; **Wichita,** LCOMP (316) 265-9507.

MARYLAND: Arrow (301) 247-5200; Diplomat (301) 995-1226; Kierulff (301) 636-5800; Milgray (301) 468-6400.

MASSACHUSETTS: Arrow (617) 933-8130; Diplomat (617) 935-6611; Kierulff (617) 667-8331; Marshall (617) 272-8200; Time (617) 935-8080.

MICHIGAN: Detroit, Arrow (313) 971-8220; Marshall (313) 525-5850; Newark (313) 967-0600; **Grand Rapids,** Arrow (616) 243-0912.

MINNESOTA: Arrow (612) 830-1800; Hall-Mark (612) 854-3223; Kierulff (612) 941-7500.

MISSOURI: Kansas City, LCOMP (816) 221-2400; **St. Louis,** Arrow (314) 567-6888; Hall-Mark (314) 291-5350; Kierulff (314) 739-0855.

NEW HAMPSHIRE: Arrow (603) 668-6968.

NEW JERSEY: Arrow (201) 575-5300, (609) 596-8000; Diplomat (201) 785-1830; General Radio (609) 964-8560; Kierulff (201) 575-6750; Marshall (201) 882-0320, (609) 234-9100; Milgray (609) 983-5010.

NEW MEXICO: Arrow (505) 243-4566; International Electronics (505) 345-8127.

NEW YORK: Long Island, Arrow (516) 231-1000; Diplomat (516) 454-6334; JACO (516) 273-5500; Marshall (516) 273-2424; Milgray (516) 420-9800; **Rochester,** Arrow (716) 275-0300; Marshall (716) 235-7620; Rochester Radio Supply (716) 454-7800; **Syracuse,** Arrow (315) 652-1000; Diplomat (315) 652-5000; Marshall (607) 754-1570.

NORTH CAROLINA: Arrow (919) 876-3132, (919) 725-8711; Kierulff (919) 872-8410.

OHIO: Cincinnati, Graham (513) 772-1661; Hall-Mark (513) 563-5980; **Cleveland,** Arrow (216) 248-3990; Hall-Mark (216) 349-4632; Kierulff (216) 587-6558; **Columbus,** Hall-Mark (614) 891-4555, **Dayton,** Arrow (513) 435-5563; ESCO (513) 226-1133; Marshall (513) 236-8088.

OKLAHOMA: Arrow (918) 665-7700; Hall-Mark (918) 665-3200; Kierulff (918) 252-7537.

OREGON: Arrow (503) 684-7690; Kierulff (503) 641-9150; Wyle (503) 640-6000.

PENNSYLVANIA: Arrow (412) 856-7000, (215) 928-1800; General Radio (215) 922-7037.

TEXAS: Austin, Arrow (512) 835-4180; Hall-Mark (512) 258-8848; Kierulff (512) 835-2090; **Dallas,** Arrow (214) 386-7500; Hall-Mark (214) 341-1147; International Electronics (214) 233-9323; Kierulff (214) 343-2400; **Wyle,** (214) 235-9953; **El Paso,** International Electronics (915) 598-3406; **Houston,** Arrow (713) 530-4700; Hall-Mark (713) 781-6100; Harrison Equipment (713) 879-2600; Kierulff (713) 530-7030.

UTAH: Diplomat (801) 486-4134; Kierulff (801) 973-6913; Wyle (801) 974-9953.

VIRGINIA: Arrow (804) 282-0413.

WASHINGTON: Arrow (206) 643-4800; Kierulff (206) 575-4420; Wyle (206) 453-8300.

WISCONSIN: Arrow (414) 764-6600; Hall-Mark (414) 761-3000; Kierulff (414) 784-8160.

CANADA: Calgary, Future (403) 486-0974; Varah (403) 230-1235; **Hamilton,** Varah (416) 561-9311; **Montreal,** CESCO (514) 735-5511; Future (514) 694-7710; ITT Components (514) 735-1177; **Ottawa,** CESCO (613) 226-6903; Future (613) 820-8313; ITT Components (613) 226-7406; Varah (613) 592-4728; **Quebec City,** CESCO (418) 687-4231; **Toronto,** CESCO (416) 661-0220; Future (416) 663-5563; ITT Components (416) 630-7971; **Vancouver,** Future (604) 438-5545; Varah (604) 873-3211; ITT Components (604) 270-7805; **Winnipeg,** Varah (204) 633-6190. BF

TEXAS INSTRUMENTS

Creating useful products
and services for you.

Printed in U.S.A.